

IEEE Standard for Device Discovery, Connection Management, and Control Protocol for IEEE 1722™ Based Devices

IEEE Computer Society

Sponsored by the
Microprocessor Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

IEEE Std 1722.1™-2013

IEEE Standard for Device Discovery, Connection Management, and Control Protocol for IEEE 1722™ Based Devices

Sponsor

Microprocessor Standards Committee
of the
IEEE Computer Society

Approved 23 August 2013

IEEE-SA Standards Board

Abstract: This standard specifies the protocol, device discovery, connection management and device control procedures used to facilitate interoperability between audio and video based End Stations that use IEEE 1722 based Streams on IEEE 802[®] based networks.

Keywords: AVDECC, bridged LAN, IEC 61883, IEEE 802.1[™] AVB protocols, IEEE 802.1BA[™], IEEE 1722.1[™], IEEE Std 802.1AS[™]-2011, IEEE Std 802.1Q[™]-2011, IEEE Std 1722[™]-2011, LAN, QoS, time sensitive media streaming, time synchronization

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2013 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 10 October 2013. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-0-7381-8624-5 STD98376
Print: ISBN 978-0-7381-8625-2 STDPD98376

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Notice and Disclaimer of Liability Concerning the Use of IEEE Documents: IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon any IEEE Standard document.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained in its standards is free from patent infringement. IEEE Standards documents are supplied “**AS IS**.”

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Translations: The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official Statements: A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on Standards: Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important to ensure that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. Any person who would like to participate in evaluating comments or revisions to an IEEE standard is welcome to join the relevant IEEE working group at <http://standards.ieee.org/develop/wg/>.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Photocopies: Authorization to photocopy portions of any individual standard for internal or personal use is granted by The Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Notice to users

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA Website at <http://standards.ieee.org/index.html> or contact the IEEE at the address listed previously. For more information about the IEEE Standards Association or the IEEE standards development process, visit IEEE-SA Website at <http://standards.ieee.org/index.html>.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: <http://standards.ieee.org/findstds/errata/index.html>. Users are encouraged to check this URL for errata periodically.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at <http://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this IEEE standard was completed, the IEEE Standard Device Discovery, Connection Management and Control Protocol for IEEE 1722 Based Devices Working Group had the following membership:

Matthew Xavier Mora, *Chair*
Michael Johas Teener, *Vice Chair*
Don Pannell, *Secretary*
Ashley Butterworth, *Co-Editor*
Jeffrey Koftinoff, *Co-Editor*

Bob Abraham
Torrey Atcitty
Eliot Blennerhassett
Robert Boatright
Philippe Boucachard
Alexander Busch
Debin Chen
George Claseman
Guy Fedorkow
Bob Feng
Richard Foss
Philip Foulkes
John Nels Fuller
Aaron Gelter

John Grant
Duncan Gray
Kevin Gross
Craig Gunther
Joern Hennenberg
Alison Hughes
Osedum P. Igumbor
Girault Jones, Jr.
Max Kicherer
Yong Kim
Matt Klein
Hans Lau
James R. Leitch
Tom Mathey
Gail McCoy

Lee Minich
David Olsen
Nathan O'Neill
Xavier Miguelez Ortiz
Chris Pane
Don Pannell
Nagaprasad Ramachandra
Vincent Rowley
Bennet Sikes
Rob Silfvast
Kevin Stanton
Lalin Theverapperuma
Stephen Turner
Niel Warren

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Thomas Alexander
Martin J. Bishop
Christian Boiger
Ashley Butterworth
Keith Chow
Rodney Cummings
James Davis
Richard Eckard
Andre Fournier
James Gilb
Patrick Gonia
Sudheer Grandhi
Randall Groves
Michael Gundlach
Craig Gunther
Jerome Henry
Marco Hernandez
Werner Hoelzl
Osedum Igumbor
Atsushi Ito
Anthony Jeffree
Michael Johas Teener

Girault Jones
Chol Kang
Piotr Karocki
Stuart Kerry
Yuri Khersonsky
Yongbum Kim
Jeff Koftinoff
Bruce Kraemer
Thomas Kurihara
Geoff Ladwig
Jan-Ray Liao
Shen Loh
Greg Luri
Jeffery Masters
Michael McInnis
Matthew Xavier Mora
Jose Morales
Michael S. Newman
Charles Ngethe
Satoshi Obara
David Olsen
Satoshi Oyama

Chris Pane
Maximilian Riegel
Jeff Rockower
Benjamin Rolfe
Dan Romascanu
Randall Safier
Peter Saunderson
Bartien Sayogo
Gil Shultz
Kevin Stanton
Walter Struppler
William Taylor
David Thompson
Kazuyoshi Tsukada
Stephen Turner
Dmitri Varsanofiev
Prabodh Varshney
John Vergis
Haiming Wang
Colin Whitby-Stevens
Forrest Wright
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 23 August 2013, it had the following membership:

John Kulick, *Chair*
David J. Law, *Vice Chair*
Richard H. Hulett, *Past Chair*
Konstantinos Karachalios, *Secretary*

Masayuki Ariyoshi
Peter Balma
Farooq Bari
Ted Burse
Wael William Diab
Stephen Dukes
Jean-Philippe Faure
Alexander Gelman

Mark Halpin
Gary Hoffman
Paul Houzé
Jim Hughes
Michael Janezic
Joseph L. Koepfinger*
David J. Law
Oleg Logvinov

Ron Petersen
Gary Robinson
Jon Walter Rosdahl
Adrian Stephens
Peter Sutherland
Yatin Trivedi
Phil Winston
Yu Yuan

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Richard DeBlasio, DOE Representative
Michael Janezic, NIST Representative

Michelle D. Turner
IEEE Standards Program Manager, Document Development

Joan Woolery
IEEE Standards Program Manager, Technical Program Development

Introduction

This introduction is not part of IEEE Std 1722.1™-2013, IEEE Standard for Device Discovery, Connection Management, and Control Protocol for IEEE 1722™ Based Devices.

Increasingly, entertainment media are digitally transported. Streaming audio/video and interactive applications over local area networks is becoming more common.

This standard builds on the work done by the IEEE 802.1™ AVB task group by providing a common audio/video transport protocol capable of supporting the needs of both consumer and professional audio/video applications.

Contents

1. Overview	1
1.1 Scope	2
1.2 Purpose	2
2. Normative references.....	3
3. Definitions, acronyms, and abbreviations.....	4
3.1 Definitions	4
3.2 Acronyms and abbreviations	6
4. Other information	8
4.1 Word usage.....	8
4.2 Numerical values	8
5. General requirements.....	8
5.1 Overview	8
5.2 AVDECC End Station	9
5.2.1 Requirements and options	10
5.3 AVDECC Entity	10
5.3.1 Requirements and options	10
5.4 AVDECC Controller	12
5.4.1 Requirements and options	12
5.4.2 Multiple Controllers	16
5.4.3 Controller behavior	16
5.5 AVDECC Talker	17
5.5.1 Requirements and options	17
5.6 AVDECC Listener.....	20
5.6.1 Requirements and options	20
5.7 AVDECC Responder.....	22
5.7.1 Requirements and options	22
5.8 AVDECC Proxy Server requirements and options.....	24
5.9 AVDECC Proxy Client requirements and options	25
6. AVDECC Entity Discovery.....	25
6.1 Overview	25
6.2 AVDECC Discovery Protocol.....	25
6.2.1 AVDECC Discovery Protocol Data Unit format.....	25
6.2.2 Protocol specification	33
6.2.3 Global state machine variables.....	33
6.2.4 Advertise Entity State Machine.....	34
6.2.5 Advertise Interface State Machine	35
6.2.6 Discovery State machine	37
7. AVDECC Entity Model.....	40
7.1 Overview	40
7.2 Descriptors.....	42
7.2.1 ENTITY Descriptor.....	44
7.2.2 CONFIGURATION Descriptor	46
7.2.3 AUDIO_UNIT Descriptor.....	47
7.2.4 VIDEO_UNIT Descriptor	49
7.2.5 SENSOR_UNIT Descriptor	51

7.2.6	STREAM_INPUT and STREAM_OUTPUT Descriptor	53
7.2.7	JACK_INPUT and JACK_OUTPUT Descriptor	55
7.2.8	AVB_INTERFACE Descriptor	57
7.2.9	CLOCK_SOURCE Descriptor	59
7.2.10	MEMORY_OBJECT Descriptor	60
7.2.11	LOCALE Descriptor	62
7.2.12	STRINGS Descriptor	63
7.2.13	STREAM_PORT_INPUT and STREAM_PORT_OUTPUT Descriptor	63
7.2.14	EXTERNAL_PORT_INPUT and EXTERNAL_PORT_OUTPUT Descriptor	65
7.2.15	INTERNAL_PORT_INPUT and INTERNAL_PORT_OUTPUT Descriptor	66
7.2.16	AUDIO_CLUSTER Descriptor	67
7.2.17	VIDEO_CLUSTER Descriptor	69
7.2.18	SENSOR_CLUSTER Descriptor	73
7.2.19	AUDIO_MAP Descriptor	75
7.2.20	VIDEO_MAP Descriptor	77
7.2.21	SENSOR_MAP Descriptor	78
7.2.22	CONTROL Descriptor	79
7.2.23	SIGNAL_SELECTOR Descriptor	82
7.2.24	MIXER Descriptor	85
7.2.25	MATRIX Descriptor	87
7.2.26	MATRIX_SIGNAL Descriptor	89
7.2.27	SIGNAL_SPLITTER Descriptor	91
7.2.28	SIGNAL_COMBINER Descriptor	93
7.2.29	SIGNAL_DEMULTIPLEXER Descriptor	96
7.2.30	SIGNAL_MULTIPLEXER Descriptor	98
7.2.31	SIGNAL_TRANSCODER Descriptor	101
7.2.32	CLOCK_DOMAIN Descriptor	104
7.2.33	CONTROL_BLOCK Descriptor	104
7.3	Descriptor Field Value Types	105
7.3.1	Sampling Rates	105
7.3.2	Stream Formats	107
7.3.3	Control Value Units	115
7.3.4	Control Types	123
7.3.5	Control Values	143
7.3.6	Localized String Reference	151
7.3.7	Video Cluster Formats Specific	151
7.3.8	Video Cluster Pixel Aspect Ratio	158
7.3.9	Video Cluster Frame Size	159
7.3.10	Video Cluster Color Space	159
7.3.11	Sensor Cluster Format	160
7.4	Commands and Responses	162
7.4.1	ACQUIRE_ENTITY Command	166
7.4.2	LOCK_ENTITY Command	168
7.4.3	ENTITY_AVAILABLE Command	169
7.4.4	CONTROLLER_AVAILABLE Command	169
7.4.5	READ_DESCRIPTOR Command	170
7.4.6	WRITE_DESCRIPTOR Command	171
7.4.7	SET_CONFIGURATION Command	172
7.4.8	GET_CONFIGURATION Command	173
7.4.9	SET_STREAM_FORMAT Command	174
7.4.10	GET_STREAM_FORMAT Command	175
7.4.11	SET_VIDEO_FORMAT Command	176
7.4.12	GET_VIDEO_FORMAT Command	177
7.4.13	SET_SENSOR_FORMAT Command	178
7.4.14	GET_SENSOR_FORMAT Command	179
7.4.15	SET_STREAM_INFO Command	180

7.4.16 GET_STREAM_INFO Command	183
7.4.17 SET_NAME Command	185
7.4.18 GET_NAME Command.....	186
7.4.19 SET_ASSOCIATION_ID Command	187
7.4.20 GET_ASSOCIATION_ID Command.....	188
7.4.21 SET_SAMPLING_RATE Command	188
7.4.22 GET_SAMPLING_RATE Command.....	189
7.4.23 SET_CLOCK_SOURCE Command.....	190
7.4.24 GET_CLOCK_SOURCE Command	191
7.4.25 SET_CONTROL Command	192
7.4.26 GET_CONTROL Command.....	193
7.4.27 INCREMENT_CONTROL Command	194
7.4.28 DECREMENT_CONTROL Command	196
7.4.29 SET_SIGNAL_SELECTOR Command.....	197
7.4.30 GET_SIGNAL_SELECTOR Command.....	198
7.4.31 SET_MIXER Command	199
7.4.32 GET_MIXER Command.....	200
7.4.33 SET_MATRIX Command	201
7.4.34 GET_MATRIX Command.....	203
7.4.35 START_STREAMING Command	205
7.4.36 STOP_STREAMING Command	206
7.4.37 REGISTER_UNSOLICITED_NOTIFICATION Command	207
7.4.38 DEREGISTER_UNSOLICITED_NOTIFICATION Command.....	207
7.4.39 IDENTIFY_NOTIFICATION Unsolicited Response.....	207
7.4.40 GET_AVB_INFO Command.....	208
7.4.41 GET_AS_PATH Command	210
7.4.42 GET_COUNTERS Command	212
7.4.43 REBOOT Command	218
7.4.44 GET_AUDIO_MAP Command	219
7.4.45 ADD_AUDIO_MAPPINGS Command.....	221
7.4.46 REMOVE_AUDIO_MAPPINGS Command.....	222
7.4.47 GET_VIDEO_MAP Command.....	222
7.4.48 ADD_VIDEO_MAPPINGS Command	224
7.4.49 REMOVE_VIDEO_MAPPINGS Command	225
7.4.50 GET_SENSOR_MAP Command.....	226
7.4.51 ADD_SENSOR_MAPPINGS Command	227
7.4.52 REMOVE_SENSOR_MAPPINGS Command	228
7.4.53 START_OPERATION Command	228
7.4.54 ABORT_OPERATION Command	230
7.4.55 OPERATION_STATUS Unsolicited Response.....	230
7.4.56 AUTH_ADD_KEY Command	231
7.4.57 AUTH_DELETE_KEY Command	233
7.4.58 AUTH_GET_KEY_LIST Command.....	233
7.4.59 AUTH_GET_KEY Command	234
7.4.60 AUTH_ADD_KEY_TO_CHAIN Command.....	235
7.4.61 AUTH_DELETE_KEY_FROM_CHAIN Command	236
7.4.62 AUTH_GET_KEYCHAIN_LIST Command	237
7.4.63 AUTH_GET_IDENTITY Command.....	238
7.4.64 AUTH_ADD_TOKEN Command	240
7.4.65 AUTH_DELETE_TOKEN Command.....	241
7.4.66 AUTHENTICATE Command.....	241
7.4.67 DEAUTHENTICATE Command	243
7.4.68 ENABLE_TRANSPORT_SECURITY Command	244
7.4.69 DISABLE_TRANSPORT_SECURITY Command	244
7.4.70 ENABLE_STREAM_ENCRYPTION Command	245
7.4.71 DISABLE_STREAM_ENCRYPTION Command	246

7.4.72 SET_MEMORY_OBJECT_LENGTH Command	247
7.4.73 GET_MEMORY_OBJECT_LENGTH Command	248
7.4.74 SET_STREAM_BACKUP Command	249
7.4.75 GET_STREAM_BACKUP Command	250
7.5 Notifications	251
7.5.1 Identification Notification	251
7.5.2 Unsolicited Notifications	253
7.6 Security	255
7.6.1 Key management	255
7.6.2 Controller Authorization	258
7.6.3 Transport Security Control	259
7.6.4 Stream Encryption Control	259
7.6.5 Entity Model Verification	259
8. Connection management	268
8.1 Overview	268
8.2 AVDECC Connection Management Protocol	268
8.2.1 AVDECC Connection Management Protocol Data Unit format	268
8.2.2 Protocol Specification	273
9. Enumeration and control	289
9.1 Overview	289
9.2 AVDECC Enumeration and Control Protocol	289
9.2.1 AVDECC Enumeration and Control Protocol Data Unit format	289
9.2.2 Protocol Operation	300
Annex A (informative) Bibliography	325
Annex B (normative) Reserved AVDECC MAC addresses	326
B.1 Overview	326
Annex C (normative) AVDECC Proxy Protocol	327
C.1 Overview	327
C.2 DNS-SD Service Name	327
C.3 DNS-SD TXT Record	328
C.4 APPDU format	328
C.5 Protocol Description	331
Annex D (informative) Memory Object Uploads	343
D.1 Overview	343
D.2 Memory Object Upload Entity State Machine	343
D.3 Memory Object Upload Controller State Machine	348
Annex E (informative) XML Representation of AVDECC Entity Models	352
E.1 Overview	352

IEEE Standard for Device Discovery, Connection Management, and Control Protocol for IEEE 1722™ Based Devices

IMPORTANT NOTICE: IEEE Standards documents are not intended to ensure safety, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading “Important Notice” or “Important Notices and Disclaimers Concerning IEEE Documents.” They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/IPR/disclaimers.html>.

1. Overview

A number of proprietary protocols exist for allowing End Stations on a local area network (LAN) to connect to each other and stream media over the network. There has been a need in the industry to come up with an industry standard way to connect and interoperate media based devices on a network. IEEE Std 1722™-2011 has defined an industry standard way to transport media Streams over a LAN and this standard will be the industry standard for connecting IEEE Std 1722-2011 streaming End Stations.

This standard, for audio/video discovery, enumeration, connection management and control (AVDECC), defines four independent steps that can be used to connect End Stations that use 1722-2011 transport Streams to transport media Streams across a LAN. The steps are as follows:

- a) Discovery
- b) Enumeration
- c) Connection management
- d) Control

These steps can be used together to form a system of End Stations that interoperate with each other in a standards compliant way. The application that will use these individual steps is called an AVDECC Controller and is the third actor in the AVDECC Talker, AVDECC Listener, and AVDECC Controller device relationship.

An AVDECC Controller may exist within an AVDECC Talker or an AVDECC Listener, or exist remotely within the network in a separate End Station. The AVDECC Controller can use the individual steps to find, connect, and control entities on the network, but it may choose to not use all of the steps if the AVDECC Controller already knows some of the information (e.g., hard coded values assigned by user/hardware switch or values from previous session establishment) that can be gained in using the steps. The only required step is connection management, because this is the step that establishes the bandwidth usage and reservations across the audio video bridging (AVB) domain.

The four steps are described as follows:

- **Discovery** is the process of finding AVDECC Entities on the LAN that have services that are useful to the other AVDECC Entities on the network. The discovery process also covers the termination of the publication of those services on the network.
- **Enumeration** is the process of collecting information from the AVDECC Entity that could help an AVDECC Controller to use the capabilities of the AVDECC Entity. This information can be used for connection management.
- **Connection management** is the process of connecting or disconnecting one or more Streams between two or more AVDECC Entities.
- **Control** is the process of adjusting a parameter on the AVDECC Entity from an AVDECC Controller. There are a number of standard types of controls used in media devices like volume control, mute control, etc. A framework of basic commands allows the control process to be extended by AVDECC Entities.

These four steps are the basis of this standard and will be described in detail in the following sections.

1.1 Scope

This standard specifies the protocol, device discovery, connection management, and device control procedures used to facilitate interoperability between audio and video based End Stations that use IEEE 1722 based Streams on IEEE 802® based networks.

1.2 Purpose

This standard will facilitate interoperability between End Stations that stream time-sensitive media and data across local area networks providing time synchronization and latency/bandwidth services. This standard defines the device discovery, connection management, Stream setup, control, and teardown protocols.

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

COLLADA-Digital Asset Schema Release 1.5.0 specification (April 2008).¹

High-bandwidth Digital Content Protection System – Interface Independent Adaptation.²

IEC 60958, Digital audio interface (all parts).

IEC 61883-4:2004, Consumer Audio/Video Equipment—Digital Interface—Part 4: MPEG2-TS Data Transmission.

IEC 61883-6:2005, Consumer Audio/Video Equipment—Digital Interface—Part 6: Audio and Music Data Transmission Protocol.

IEC 61883-8:2008, Consumer Audio/Video Equipment—Digital Interface—Part 8: Transmission of ITU-R BT.601 Style Digital Video Data.

IEEE Std 754™-2008, IEEE Standard for Floating Point Arithmetic.^{3, 4}

IEEE Std 802.1AS™-2011, IEEE Standard for Local and Metropolitan Area Networks—Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks.

IEEE Std 802.1BA™-2011, IEEE Standard for Local and Metropolitan Area Networks—Audio Video Bridging (AVB) Systems.

IEEE Std 802.1Q™-2011, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks.

IEEE Std 1363a™-2004, Standard Specifications For Public-Key Cryptography—Amendment 1: Additional Techniques.

IEEE Std 1541™-2002, IEEE Standard for Prefixes for Binary Multiples.

IEEE Std 1722™-2011, IEEE Standard for Layer 2 Transport Protocol for Time Sensitive Applications in a Bridged Local Area Network.

RFC 2435, RTP Payload Format for JPEG-compressed Video.⁵

RFC 2616, Hypertext Transfer Protocol—HTTP/1.1.⁶

RFC 5371, RTP Payload Format for JPEG 2000 Video Streams.⁷

RFC 6184, RTP Payload Format for H.264 Video.⁸

W3C Portable Network Graphics (PNG) Specification (Second Edition).⁹

W3C Scalable Vector Graphics (SVG) 1.1 (Second Edition).¹⁰

¹ Available at http://www.khronos.org/files/collada_spec_1_5.pdf

² Available at http://www.digital-cp.com/files/static_page_files/DABB540C-1A4B-B294-D0008CB2D348FA19/HDCP%20Interface%20Independent%20Adaptation%20Specification%20Rev2_1.pdf

³ IEEE publications are available from The Institute of Electrical and Electronics Engineers (<http://standards.ieee.org/>).

⁴ The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.

⁵ Available at <http://tools.ietf.org/html/rfc2435>

⁶ Available at <http://tools.ietf.org/html/rfc2616>

⁷ Available at <http://tools.ietf.org/html/rfc5371>

⁸ Available at <http://tools.ietf.org/html/rfc6184>

⁹ Available at <http://www.w3.org/TR/PNG/>

¹⁰ Available at <http://www.w3.org/TR/SVG/>

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.¹¹

Active Stream: A data Stream associated with a stream reservation established using the Stream Reservation Protocol (SRP), which has data flowing from a Talker to one or more Listeners.

ADAT® Optical Interface: The Alesis Digital Audio Tape Optical Interface digital audio interconnect.^{12, 13}

Audio Cluster: A group of audio channels in a stream.

Audio Map: A static mapping between an audio Stream's channels and an Audio Cluster's channels for Streams and Stream Ports.

Audio Unit: A Unit for processing audio.

AVB domain: See IEEE Std 802.1BA™.

AVB Interface: An AVB capable network interface capable of sourcing or sinking Streams and participating in an AVB domain.

AVDECC Controller: Any entity defined by this standard that initiates exchanges with any other entity defined by this standard.

AVDECC Entity: A logical object within an End Station that can accept and respond to AVDECC messages.

AVDECC Entity Model: A collection of objects that define the internal structure of the AVDECC Entity and the flow of media through it.

AVDECC Listener: An AVDECC Entity that can transmit or receive the AVDECC protocol and is an AVTP Listener.

AVDECC Proxy Client (APC): An End Station that implements the AVDECC Proxy Client state machine.

AVDECC Proxy Protocol (APP): The TCP protocol used to transport AVDECC messages between an AVDECC Proxy Client and an AVDECC Proxy Server.

AVDECC Proxy Server (APS): An End Station that implements the AVDECC Proxy Server state machine.

AVDECC Talker: An AVDECC Entity that can transmit or receive the AVDECC protocol and is an AVTP Talker.

AVTP End Station: An End Station that can transmit or receive the IEEE Std 1722™ protocol.

¹¹ *IEEE Standards Dictionary Online* subscription is available at:
http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html.

¹² ADAT® is a registered trademark owned by ALESIS CORPORATION CORPORATION CALIFORNIA 3630 Holdrege Avenue Los Angeles CALIFORNIA 90016.

¹³ This information is given for the convenience of users of this standard and does not constitute an endorsement by IEEE of these products. Equivalent products may be used if they can be shown to lead to the same results.

AVTP gateway: A device that transports AVTP audio/video Streams between an AVB network and another type of media (e.g., an IEEE Std 1394™ bus).

AVTP Listener: An End Station that implements a Listener as defined in IEEE Std 1722™.

AVTP Talker: An End Station that implements a Talker as defined in IEEE Std 1722™.

Clock Domain: A source of a common clock signal within an AVDECC Entity.

Clock Source: A source of timing information for a Clock Domain.

Configuration: A collection of objects that defines the internal structure for a particular setup of the AVDECC Entity.

Control: A collection of descriptive meta-data and associated values for an observable and/or changeable object within the AVDECC Entity representing a processing element or a transducer.

Control Block: A grouping of Controls.

End Station: See IEEE Std 802®.

Entity ID: The EUI-64 identifier of an AVDECC Entity.

External Port: A Port that connects to a Jack.

Internal Port: A Port that is connected to another Internal Port within an AVDECC Entity.

Jack: An ingress or egress point of a non-media-stream signal to or from an AVDECC Entity.

Matrix: A collection of orthogonal Controls arranged in a two dimensional array.

Media component: Fundamental data within an IEEE Std 1722™ stream payload.

Memory Object: An addressable region of memory within an AVDECC Entity.

Mixer: A Control that combines multiple signals into a single output signal.

Octlet: Eight octets of data.

Port: An ingress and egress point of a signal for a Unit.

Presentation time: A 32-bit unsigned field in units of 1 nanosecond.

Presentation time offset: A 64-bit signed field that defines an offset time in nanoseconds.

Reserved field: A data field that is reserved for future standardization and is set to 0 when transmitted and ignored when received.

Sensor Cluster: An element of a Sensor Stream.

Sensor Map: A mapping between a Sensor Stream's components and the Sensor Clusters for Streams and Stream Ports that are located in the same Clock Domain.

Sensor Unit: A Unit for processing Sensors.

Signal Combiner: A Control for combining several signals into a single signal.

Signal Demultiplexer: A Control for demultiplexing a signal into multiple signals.

Signal Multiplexer: A Control for multiplexing multiple signals into a single signal.

Signal Selector: A Control for switchable signal routing.

Signal Splitter: A Control for splitting a signal into multiple sub-signals.

Signal Transcoder: A Control for transcoding from one unencoded or encoded signal to another encoded or unencoded signal.

Stream: A unidirectional flow of IEEE Std 1722™ frames with the same StreamID.

Stream Port: A Port that is connected to a Stream.

Stream sink: A component within an AVDECC Entity that consumes a Stream.

Stream source: A component within an AVDECC Entity that produces a Stream.

Unit: Processing for single Clock Domain.

Video Cluster: An element of a video Stream.

Video Map: A static mapping between a video Stream's components and the Video Clusters for Streams and Stream Ports.

Video Unit: A Unit for processing Video.

3.2 Acronyms and abbreviations

ACMP	AVDECC Connection Management Protocol
ACMPDU	AVDECC Connection Management Protocol Data Unit
ADP	AVDECC Discovery Protocol
ADPDU	AVDECC Discovery Protocol Data Unit
AECp	AVDECC Enumeration and Control Protocol
AECpDU	AVDECC Enumeration and Control Protocol Data Unit
AEM	AVDECC Entity Model
APC	AVDECC Proxy Client
APP	AVDECC Proxy Protocol
APS	AVDECC Proxy Server
ASCII	American Standard Code for Information Interchange

AVB	Audio/video bridging
AVDECC	Audio/video discovery, enumeration, connection management, and control
AVTP	Audio/video transport protocol
AVTPDU	Audio/video transport protocol data unit
EUI	IEEE Extended Unique Identifier
FIPS	Federal Information Processing Standards ¹⁴
gPTP	Generalized Precision Time Protocol
LAN	Local Area Network
MAAP	MAC address acquisition protocol
MAC	Media access control
MIDI	Musical Instrument Digital Interface ¹⁵
MPEG	Moving Pictures Expert Group ¹⁶
MSRP	Multiple Stream Reservation Protocol
OUI	IEEE organizationally unique identifier
PDU	Protocol Data Unit
S/PDIF:	The Sony/Philips Digital Interconnect Format digital audio interconnect.
SMPTE	Society of Motion Picture and Television Engineers ¹⁷
SRP	Stream Reservation Protocol
TCP	Transmission Control Protocol ¹⁸
VLAN	Virtual Local Area Network

¹⁴ <http://itl.nist.gov/fipspubs>

¹⁵ <http://www.midi.org/>

¹⁶ <http://www.chiariglione.org/mpeg/>

¹⁷ <http://www.smpite.org/>

¹⁸ <http://tools.ietf.org/html/rfc1122>

4. Other information

4.1 Word usage

In this document, the word *shall* is used to indicate a mandatory requirement. The word *should* is used to indicate a recommendation. The word *may* is used to indicate a permissible action. The word *can* is used for statements of possibility and capability.

4.2 Numerical values

Decimal, hexadecimal, and binary numbers are used within this standard. For clarity, decimal numbers are generally used to represent counts, hexadecimal numbers are used to represent addresses, and binary numbers are used to describe bit patterns within binary fields.

Decimal numbers are represented in their usual 0, 1, 2, ... format. Hexadecimal numbers are represented by a string of one or more hexadecimal (0 to 9, a to f) digits followed by the subscript "16." Binary numbers are represented by a string of one or more binary (0, 1) digits in left to right order, where the leftmost bit is the most significant bit and the rightmost bit is the least significant bit, followed by the subscript "2." Thus, the decimal number "26" may also be represented as "1a₁₆" or "11010₂".

The notational conventions have the following exceptions:

- MAC addresses and OUI/EUI values¹⁹ are represented as strings of 8-bit hexadecimal numbers separated by hyphens and without a subscript, as for example "90-e0-f0-00-00-00-00-15", "90-e0-f0-00-00-15" or "90-e0-f0".

5. General requirements

5.1 Overview

Audio/video discovery, enumeration, connection management, and control (AVDECC) defines high-level communication protocols between AVDECC Entities, which allows them to discover other AVDECC Entities, enumerate their capabilities, manage audio/video bridging (AVB) streams, control media parameters, and allows AVDECC Entities that encounter errors to report their health and diagnostic status to AVDECC Controllers.

This standard defines four roles that an AVDECC Entity may implement:

- AVDECC Controller
- AVDECC Talker
- AVDECC Listener
- AVDECC Responder

¹⁹ Administration of standard OUI/EUI address assignments is the responsibility of the IEEE Registration Authority (IEEE RA). A full list of standard assignments, and the criteria for assignment, can be found on the IEEE RA website at <http://standards.ieee.org/develop/regauth/>.

5.2 AVDECC End Station

An AVDECC End Station is a device that has one or more network ports and has one or more AVDECC Entities.

Figure 5.1 shows the possible interaction of the various parts of an AVDECC End Station.

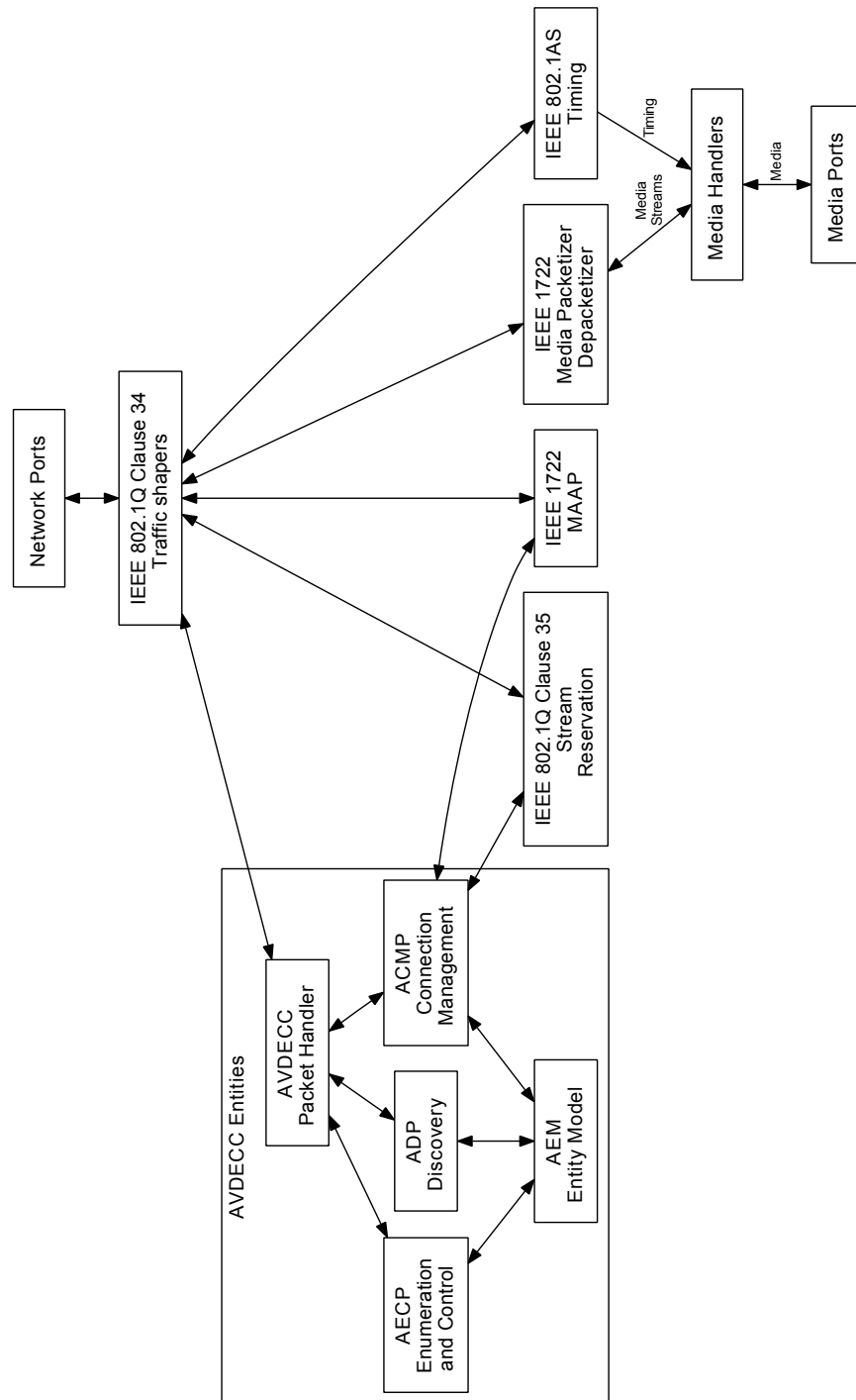


Figure 5.1—AVDECC End Station

5.2.1 Requirements and options

An AVDECC End Station shall implement:

- At least one AVDECC Entity
- At least one network interface
- IEEE 1722 Control Audio/Video Transport Protocol (AVTP) Data Unit (AVTPDU) packetization and depacketization

An AVDECC End Station may implement:

- Multiple AVDECC Entities
- One or more network interfaces that implement an appropriate AVB profile as defined in IEEE Std 802.1BA-2011
- IEEE 802.1AS time synchronization
- IEEE 1722 Stream AVTPDU packetization
- IEEE 1722 Stream AVTPDU depacketization
- IEEE 1722 Multicast Address Allocation Protocol (MAAP)
- IEEE 802.1Q Clause 34 FQTS Traffic Shaping
- IEEE 802.1Q Clause 35 Stream Reservation Protocol

5.3 AVDECC Entity

An AVDECC Entity uses one or more of the following AVDECC protocols for discovering or controlling other AVDECC Entities or for being discovered or controlled by other AVDECC Entities:

- AVDECC Discovery Protocol (ADP)
- AVDECC Connection Management Protocol (ACMP)
- AVDECC Enumeration and Control Protocol (AECp)

5.3.1 Requirements and options

An AVDECC Entity shall:

- Have a single Entity ID that is assigned by the manufacturer or via an AVDECC Proxy Server.
- Implement at least one of the AVDECC Controller, AVDECC Talker, AVDECC Listener, or the AVDECC Responder role.

A discoverable AVDECC Entity is an AVDECC Entity that is capable of advertising itself on a local area network (LAN). An AVDECC Entity that is to be discoverable on a network shall implement:

- 6.2.4 “Advertise Entity State ”
- 6.2.5 “Advertise Interface State ”

An AVDECC Entity that is to be discoverable on a network may implement:

— 7.5.1 “Identification Notification”

An AVDECC Entity that has a need to discover other AVDECC Entities on a LAN shall implement:

— 6.2.6 “Discovery State machine”

An AVDECC Entity that implements 9.2.2.3.1 “AVDECC Entity Model Entity State Machine” that implements 7.4.5 “READ_DESCRIPTOR Command” shall implement:

— 7.2.1 “ENTITY Descriptor”

An AVDECC Entity that implements 9.2.2.3.1 “AVDECC Entity Model Entity State Machine” that implements 7.4.5 “READ_DESCRIPTOR Command” may implement:

— 7.2.2 “CONFIGURATION Descriptor”

— 7.2.3 “AUDIO_UNIT Descriptor”

— 7.2.4 “VIDEO_UNIT Descriptor”

— 7.2.5 “SENSOR_UNIT Descriptor”

— 7.2.6 “STREAM_INPUT and STREAM_OUTPUT Descriptor”

— 7.2.7 “JACK_INPUT and JACK_OUTPUT Descriptor”

— 7.2.8 “AVB_INTERFACE Descriptor”

— 7.2.9 “CLOCK_SOURCE Descriptor”

— 7.2.10 “MEMORY_OBJECT Descriptor”

— 7.2.11 “LOCALE Descriptor”

— 7.2.12 “STRINGS Descriptor”

— 7.2.13 “STREAM_PORT_INPUT and STREAM_PORT_OUTPUT Descriptor”

— 7.2.14 “EXTERNAL_PORT_INPUT and EXTERNAL_PORT_OUTPUT Descriptor”

— 7.2.15 “INTERNAL_PORT_INPUT and INTERNAL_PORT_OUTPUT Descriptor”

— 7.2.16 “AUDIO_CLUSTER Descriptor”

— 7.2.17 “VIDEO_CLUSTER Descriptor”

— 7.2.18 “SENSOR_CLUSTER Descriptor”

— 7.2.19 “AUDIO_MAP Descriptor”

— 7.2.20 “VIDEO_MAP Descriptor”

— 7.2.21 “SENSOR_MAP Descriptor”

— 7.2.22 “CONTROL Descriptor”

— 7.2.23 “SIGNAL_SELECTOR Descriptor”

— 7.2.24 “MIXER Descriptor”

— 7.2.25 “MATRIX Descriptor”

— 7.2.26 “MATRIX_SIGNAL Descriptor”

— 7.2.27 “SIGNAL_SPLITTER Descriptor”

— 7.2.28 “SIGNAL_COMBINER Descriptor”

- 7.2.29 “SIGNAL_DEMULTIPLEXER Descriptor”
- 7.2.30 “SIGNAL_MULTIPLEXER Descriptor”
- 7.2.31 “SIGNAL_TRANSCODER Descriptor”
- 7.2.32 “CLOCK_DOMAIN Descriptor”
- 7.2.33 “CONTROL_BLOCK Descriptor”

5.4 AVDECC Controller

An AVDECC Entity that has the role of AVDECC Controller initiates commands to and receives responses from other AVDECC Entities.

5.4.1 Requirements and options

An AVDECC Controller shall use AVDECC messages transported via IEEE 1722 AVTPDUs or via an AVDECC Proxy Server (APS).

An AVDECC Controller shall implement:

- 9.2.1.2 “AVDECC Entity Model format”
- 9.2.2.1 “AVDECC Entity Model Commands”
- 9.2.2.2 “AVDECC Entity Model Responses”
- 9.2.2.3.2 “AVDECC Entity Model Controller State Machine,” implementing the following:
 - 7.4.1 “ACQUIRE_ENTITY Command”
 - 7.4.2 “LOCK_ENTITY Command”
 - 7.4.3 “ENTITY_AVAILABLE Command”
 - 7.4.4 “CONTROLLER_AVAILABLE Command”

An AVDECC Controller may implement:

- 8.2.2.4 “ACMP Controller state machine”
- 9.2.2.3.2 “AVDECC Entity Model Controller State Machine”, implementing zero or more of the following:
 - 7.4.5 “READ_DESCRIPTOR Command”
 - 7.4.6 “WRITE_DESCRIPTOR Command”
 - 7.4.7 “SET_CONFIGURATION Command”
 - 7.4.8 “GET_CONFIGURATION Command”
 - 7.4.9 “SET_STREAM_FORMAT Command”
 - 7.4.10 “GET_STREAM_FORMAT Command”
 - 7.4.11 “SET_VIDEO_FORMAT Command”
 - 7.4.12 “GET_VIDEO_FORMAT Command”
 - 7.4.13 “SET_SENSOR_FORMAT Command”

- 7.4.14 “GET_SENSOR_FORMAT Command”
- 7.4.15 “SET_STREAM_INFO Command”
- 7.4.16 “GET_STREAM_INFO Command”
- 7.4.17 “SET_NAME Command”
- 7.4.18 “GET_NAME Command”
- 7.4.19 “SET_ASSOCIATION_ID Command”
- 7.4.20 “GET_ASSOCIATION_ID Command”
- 7.4.21 “SET_SAMPLING_RATE Command”
- 7.4.22 “GET_SAMPLING_RATE Command”
- 7.4.23 “SET_CLOCK_SOURCE Command”
- 7.4.24 “GET_CLOCK_SOURCE Command”
- 7.4.25 “SET_CONTROL Command”
- 7.4.26 “GET_CONTROL Command”
- 7.4.27 “INCREMENT_CONTROL Command”
- 7.4.28 “DECREMENT_CONTROL Command”
- 7.4.29 “SET_SIGNAL_SELECTOR Command”
- 7.4.30 “GET_SIGNAL_SELECTOR Command”
- 7.4.31 “SET_MIXER Command”
- 7.4.32 “GET_MIXER Command”
- 7.4.33 “SET_MATRIX Command”
- 7.4.34 “GET_MATRIX Command”
- 7.4.35 “START_STREAMING Command”
- 7.4.36 “STOP_STREAMING Command”
- 7.4.37 “REGISTER_UNSOLICITED_NOTIFICATION Command”
- 7.4.38 “DEREGISTER_UNSOLICITED_NOTIFICATION Command”
- 7.4.39 “IDENTIFY_NOTIFICATION Unsolicited Response”
- 7.4.40 “GET_AVB_INFO Command”
- 7.4.41 “GET_AS_PATH Command”
- 7.4.42 “GET_COUNTERS Command”
- 7.4.43 “REBOOT Command”
- 7.4.44 “GET_AUDIO_MAP Command”
- 7.4.45 “ADD_AUDIO_MAPPINGS Command”
- 7.4.46 “REMOVE_AUDIO_MAPPINGS Command”
- 7.4.47 “GET_VIDEO_MAP Command”
- 7.4.48 “ADD_VIDEO_MAPPINGS Command”

- 7.4.49 “REMOVE_VIDEO_MAPPINGS Command”
- 7.4.50 “GET_SENSOR_MAP Command”
- 7.4.51 “ADD_SENSOR_MAPPINGS Command”
- 7.4.52 “REMOVE_SENSOR_MAPPINGS Command”
- 7.4.53 “START_OPERATION Command”
- 7.4.54 “ABORT_OPERATION Command”
- 7.4.55 “OPERATION_STATUS Unsolicited Response”
- 7.4.56 “AUTH_ADD_KEY Command”
- 7.4.57 “AUTH_DELETE_KEY Command”
- 7.4.58 “AUTH_GET_KEY_LIST Command”
- 7.4.59 “AUTH_GET_KEY Command”
- 7.4.60 “AUTH_ADD_KEY_TO_CHAIN Command”
- 7.4.61 “AUTH_DELETE_KEY_FROM_CHAIN Command”
- 7.4.62 “AUTH_GET_KEYCHAIN_LIST Command”
- 7.4.63 “AUTH_GET_IDENTITY Command”
- 7.4.64 “AUTH_ADD_TOKEN Command”
- 7.4.65 “AUTH_DELETE_TOKEN Command”
- 7.4.66 “AUTHENTICATE Command”
- 7.4.67 “DEAUTHENTICATE Command”
- 7.4.68 “ENABLE_TRANSPORT_SECURITY Command”
- 7.4.69 “DISABLE_TRANSPORT_SECURITY Command”
- 7.4.70 “ENABLE_STREAM_ENCRYPTION Command”
- 7.4.71 “DISABLE_STREAM_ENCRYPTION Command”
- 7.4.72 “SET_MEMORY_OBJECT_LENGTH Command”
- 7.4.73 “GET_MEMORY_OBJECT_LENGTH Command”
- 7.4.74 “SET_STREAM_BACKUP Command”
- 7.4.75 “GET_STREAM_BACKUP Command”
- 9.2.2.3.1 “AVDECC Entity Model Entity State Machine,” implementing zero or more of the following:
 - 7.4.1 “ACQUIRE_ENTITY Command”
 - 7.4.2 “LOCK_ENTITY Command”
 - 7.4.3 “ENTITY_AVAILABLE Command”
 - 7.4.4 “CONTROLLER_AVAILABLE Command”
 - 7.4.5 “READ_DESCRIPTOR Command”
 - 7.4.6 “WRITE_DESCRIPTOR Command”
 - 7.4.7 “SET_CONFIGURATION Command”

- 7.4.8 “GET_CONFIGURATION Command”
- 7.4.17 “SET_NAME Command”
- 7.4.18 “GET_NAME Command”
- 7.4.19 “SET_ASSOCIATION_ID Command”
- 7.4.20 “GET_ASSOCIATION_ID Command”
- 7.4.21 “SET_SAMPLING_RATE Command”
- 7.4.22 “GET_SAMPLING_RATE Command”
- 7.4.23 “SET_CLOCK_SOURCE Command”
- 7.4.24 “GET_CLOCK_SOURCE Command”
- 7.4.25 “SET_CONTROL Command”
- 7.4.26 “GET_CONTROL Command”
- 7.4.27 “INCREMENT_CONTROL Command”
- 7.4.28 “DECREMENT_CONTROL Command”
- 7.4.29 “SET_SIGNAL_SELECTOR Command”
- 7.4.30 “GET_SIGNAL_SELECTOR Command”
- 7.4.31 “SET_MIXER Command”
- 7.4.32 “GET_MIXER Command”
- 7.4.33 “SET_MATRIX Command”
- 7.4.34 “GET_MATRIX Command”
- 7.4.37 “REGISTER_UNSOLICITED_NOTIFICATION Command”
- 7.4.38 “DEREGISTER_UNSOLICITED_NOTIFICATION Command”
- 7.4.39 “IDENTIFY_NOTIFICATION Unsolicited Response”
- 7.4.40 “GET_AVB_INFO Command”
- 7.4.41 “GET_AS_PATH Command”
- 7.4.42 “GET_COUNTERS Command”
- 7.4.43 “REBOOT Command”
- 7.4.53 “START_OPERATION Command”
- 7.4.54 “ABORT_OPERATION Command”
- 7.4.55 “OPERATION_STATUS Unsolicited Response”
- 7.4.56 “AUTH_ADD_KEY Command”
- 7.4.57 “AUTH_DELETE_KEY Command”
- 7.4.58 “AUTH_GET_KEY_LIST Command”
- 7.4.59 “AUTH_GET_KEY Command”
- 7.4.60 “AUTH_ADD_KEY_TO_CHAIN Command”
- 7.4.61 “AUTH_DELETE_KEY_FROM_CHAIN Command”

- 7.4.62 “AUTH_GET_KEYCHAIN_LIST Command”
- 7.4.63 “AUTH_GET_IDENTITY Command”
- 7.4.64 “AUTH_ADD_TOKEN Command”
- 7.4.65 “AUTH_DELETE_TOKEN Command”
- 7.4.66 “AUTHENTICATE Command”
- 7.4.67 “DEAUTHENTICATE Command”
- 7.4.68 “ENABLE_TRANSPORT_SECURITY Command”
- 7.4.69 “DISABLE_TRANSPORT_SECURITY Command”
- 7.4.72 “SET_MEMORY_OBJECT_LENGTH Command”
- 7.4.73 “GET_MEMORY_OBJECT_LENGTH Command”
- 7.4.74 “SET_STREAM_BACKUP Command”
- 7.4.75 “GET_STREAM_BACKUP Command”
- 9.2.2.6.2 “Address Access Controller State Machine”
- 9.2.2.9.2 “Legacy AV/C Controller State Machine”
- 9.2.2.12.2 “Vendor Unique Controller State Machine”

5.4.2 Multiple Controllers

There may be multiple AVDECC Controllers on a single LAN at the same time. These Controllers may discover and interact with any AVDECC entities on the network.

When an AVDECC Controller needs exclusive control of an AVDECC Entity for changing or limiting changing the state of the AVDECC Entity, the AVDECC Controller uses the ACQUIRE_ENTITY or LOCK_ENTITY AVDECC Entity Model (AEM) AECp commands. While an AVDECC Entity has been acquired with the ACQUIRE_ENTITY command or locked with the LOCK_ENTITY command, another AVDECC Controller is not allowed to perform any modifications to the AVDECC Entity, however it may read values or descriptors from the AVDECC Entity Model. Typically, an AVDECC Controller would only use the LOCK_ENTITY command on another AVDECC Entity for the time needed to complete a single atomic operation, whereas it would use the ACQUIRE_ENTITY command on the AVDECC Entity for long-term exclusive control of the AVDECC Entity.

5.4.3 Controller behavior

5.4.3.1 How to handle gPTP Domain and SRP Domain changes

When the AVDECC Controller itself is participating in an AVB domain and is actively filtering entities that are in that AVB domain, and the generalized precision time protocol (gPTP, IEEE Std 802.1AS-2011) grandmaster of that AVB domain changes, then the AVDECC Controller needs to make a decision on what Entities are still in the same AVB domain as the AVDECC Controller since the grandmaster change may have been triggered by a network topology change that resulted in one or more Entities being removed from the AVB domain.

This decision seems trivial on the surface, since it applies the same check used to initially determine if the AVDECC Entity was in the same AVB domain, which is to look at the grandmaster ID in the discovery process. However, it is not as simple as that, since it takes time for the discovery process to propagate the

grandmaster change. The AVDECC Controller shall therefore wait a period of time after it has finished running its gPTP Best Master Clock Algorithm before deciding that an AVDECC Entity is no longer in its gPTP domain. It is recommended that the delay be a minimum of 2 seconds, although this can still cause an AVDECC Entity to be falsely removed if it has a large time between announces and the first announce frame after the grandmaster changeover is lost.

The same concept applies to the SR Class A and Class B VLAN ID and Priority Code Point values. The propagation of the Stream Reservation Protocol (SRP) domain change is not instantaneous across the network and could theoretically take up to 30 seconds to occur since it may have to wait up to two leave all periods of the SRP timers. In this case, the AVDECC Controller shall wait at least 30 seconds before responding to the SRP domain information change.

5.4.3.2 Entity discovery value changes

Any Controller that is determining which Entities are in the same gPTP and SRP domains needs to take into account transitory periods during which the gPTP domain or SRP domain may be in flux.

It is recommended that any Controller wait at least 2 seconds after seeing a change in the attributes of an already discovered device before acting on that change.

5.5 AVDECC Talker

An AVDECC Talker is an AVDECC Entity that can source one or more AVTP Streams.

5.5.1 Requirements and options

An AVDECC Talker shall use AVDECC messages transported via IEEE 1722 AVTPDUs.

An AVDECC Talker shall implement the following:

- IEEE 1722 AVTP Talker
- 8.2.2.6 “ACMP Talker State Machine”
- 9.2.1.2 “AVDECC Entity Model format”
- 9.2.2.1 “AVDECC Entity Model Commands”
- 9.2.2.2 “AVDECC Entity Model Responses”
- 9.2.2.3.1 “AVDECC Entity Model Entity State Machine,” implementing the following:
 - 7.4.1 “ACQUIRE_ENTITY Command”
 - 7.4.2 “LOCK_ENTITY Command”
 - 7.4.3 “ENTITY_AVAILABLE Command”
 - 7.4.4 “CONTROLLER_AVAILABLE Command”

An AVDECC Talker may implement the following:

- 9.2.2.3.1 “AVDECC Entity Model Entity State Machine,” implementing zero or more of the following:
 - 7.4.5 “READ_DESCRIPTOR Command”

- 7.4.6 “WRITE_DESCRIPTOR Command”
- 7.4.7 “SET_CONFIGURATION Command”
- 7.4.8 “GET_CONFIGURATION Command”
- 7.4.9 “SET_STREAM_FORMAT Command”
- 7.4.10 “GET_STREAM_FORMAT Command”
- 7.4.11 “SET_VIDEO_FORMAT Command”
- 7.4.12 “GET_VIDEO_FORMAT Command”
- 7.4.13 “SET_SENSOR_FORMAT Command”
- 7.4.14 “GET_SENSOR_FORMAT Command”
- 7.4.15 “SET_STREAM_INFO Command”
- 7.4.16 “GET_STREAM_INFO Command”
- 7.4.17 “SET_NAME Command”
- 7.4.18 “GET_NAME Command”
- 7.4.19 “SET_ASSOCIATION_ID Command”
- 7.4.20 “GET_ASSOCIATION_ID Command”
- 7.4.21 “SET_SAMPLING_RATE Command”
- 7.4.22 “GET_SAMPLING_RATE Command”
- 7.4.23 “SET_CLOCK_SOURCE Command”
- 7.4.24 “GET_CLOCK_SOURCE Command”
- 7.4.25 “SET_CONTROL Command”
- 7.4.26 “GET_CONTROL Command”
- 7.4.27 “INCREMENT_CONTROL Command”
- 7.4.28 “DECREMENT_CONTROL Command”
- 7.4.29 “SET_SIGNAL_SELECTOR Command”
- 7.4.30 “GET_SIGNAL_SELECTOR Command”
- 7.4.31 “SET_MIXER Command”
- 7.4.32 “GET_MIXER Command”
- 7.4.33 “SET_MATRIX Command”
- 7.4.34 “GET_MATRIX Command”
- 7.4.35 “START_STREAMING Command”
- 7.4.36 “STOP_STREAMING Command”
- 7.4.37 “REGISTER_UNSOLICITED_NOTIFICATION Command”
- 7.4.38 “DEREGISTER_UNSOLICITED_NOTIFICATION Command”
- 7.4.39 “IDENTIFY_NOTIFICATION Unsolicited Response”
- 7.4.40 “GET_AVB_INFO Command”

- 7.4.41 “GET_AS_PATH Command”
- 7.4.42 “GET_COUNTERS Command”
- 7.4.43 “REBOOT Command”
- 7.4.44 “GET_AUDIO_MAP Command”
- 7.4.45 “ADD_AUDIO_MAPPINGS Command”
- 7.4.46 “REMOVE_AUDIO_MAPPINGS Command”
- 7.4.47 “GET_VIDEO_MAP Command”
- 7.4.48 “ADD_VIDEO_MAPPINGS Command”
- 7.4.49 “REMOVE_VIDEO_MAPPINGS Command”
- 7.4.50 “GET_SENSOR_MAP Command”
- 7.4.51 “ADD_SENSOR_MAPPINGS Command”
- 7.4.52 “REMOVE_SENSOR_MAPPINGS Command”
- 7.4.53 “START_OPERATION Command”
- 7.4.54 “ABORT_OPERATION Command”
- 7.4.55 “OPERATION_STATUS Unsolicited Response”
- 7.4.56 “AUTH_ADD_KEY Command”
- 7.4.57 “AUTH_DELETE_KEY Command”
- 7.4.58 “AUTH_GET_KEY_LIST Command”
- 7.4.59 “AUTH_GET_KEY Command”
- 7.4.60 “AUTH_ADD_KEY_TO_CHAIN Command”
- 7.4.61 “AUTH_DELETE_KEY_FROM_CHAIN Command”
- 7.4.62 “AUTH_GET_KEYCHAIN_LIST Command”
- 7.4.63 “AUTH_GET_IDENTITY Command”
- 7.4.64 “AUTH_ADD_TOKEN Command”
- 7.4.65 “AUTH_DELETE_TOKEN Command”
- 7.4.66 “AUTHENTICATE Command”
- 7.4.67 “DEAUTHENTICATE Command”
- 7.4.68 “ENABLE_TRANSPORT_SECURITY Command”
- 7.4.69 “DISABLE_TRANSPORT_SECURITY Command”
- 7.4.70 “ENABLE_STREAM_ENCRYPTION Command”
- 7.4.71 “DISABLE_STREAM_ENCRYPTION Command”
- 7.4.72 “SET_MEMORY_OBJECT_LENGTH Command”
- 7.4.73 “GET_MEMORY_OBJECT_LENGTH Command”
- 7.4.74 “SET_STREAM_BACKUP Command”
- 7.4.75 “GET_STREAM_BACKUP Command”

- 9.2.2.6.1 “Address Access Entity State Machine”
- 9.2.2.9.1 “Legacy A/VC Entity State Machine”
- 9.2.2.12.1 “Vendor Unique Entity State Machine”

5.6 AVDECC Listener

An AVDECC Listener is an AVDECC Entity that can sink one or more AVTP Streams.

5.6.1 Requirements and options

An AVDECC Listener shall use AVDECC messages transported via IEEE 1722 AVTPDUs.

An AVDECC Listener shall implement the following:

- IEEE 1722 AVTP Listener
- 8.2.2.5 “ACMP Listener State Machine”
- 9.2.1.2 “AVDECC Entity Model format”
- 9.2.2.1 “AVDECC Entity Model Commands”
- 9.2.2.2 “AVDECC Entity Model Responses”
- 9.2.2.3.1 “AVDECC Entity Model Entity State Machine,” implementing the following:
 - 7.4.1 “ACQUIRE_ENTITY Command”
 - 7.4.2 “LOCK_ENTITY Command”
 - 7.4.3 “ENTITY_AVAILABLE Command”
 - 7.4.4 “CONTROLLER_AVAILABLE Command”

An AVDECC Listener may implement:

- 9.2.2.3.1 “AVDECC Entity Model Entity State Machine,” implementing zero or more of the following:
 - 7.4.5 “READ_DESCRIPTOR Command”
 - 7.4.6 “WRITE_DESCRIPTOR Command”
 - 7.4.7 “SET_CONFIGURATION Command”
 - 7.4.8 “GET_CONFIGURATION Command”
 - 7.4.9 “SET_STREAM_FORMAT Command”
 - 7.4.10 “GET_STREAM_FORMAT Command”
 - 7.4.11 “SET_VIDEO_FORMAT Command”
 - 7.4.12 “GET_VIDEO_FORMAT Command”
 - 7.4.13 “SET_SENSOR_FORMAT Command”
 - 7.4.14 “GET_SENSOR_FORMAT Command”
 - 7.4.15 “SET_STREAM_INFO Command”

- 7.4.16 “GET_STREAM_INFO Command”
- 7.4.17 “SET_NAME Command”
- 7.4.18 “GET_NAME Command”
- 7.4.19 “SET_ASSOCIATION_ID Command”
- 7.4.20 “GET_ASSOCIATION_ID Command”
- 7.4.21 “SET_SAMPLING_RATE Command”
- 7.4.22 “GET_SAMPLING_RATE Command”
- 7.4.23 “SET_CLOCK_SOURCE Command”
- 7.4.24 “GET_CLOCK_SOURCE Command”
- 7.4.25 “SET_CONTROL Command”
- 7.4.26 “GET_CONTROL Command”
- 7.4.27 “INCREMENT_CONTROL Command”
- 7.4.28 “DECREMENT_CONTROL Command”
- 7.4.29 “SET_SIGNAL_SELECTOR Command”
- 7.4.30 “GET_SIGNAL_SELECTOR Command”
- 7.4.31 “SET_MIXER Command”
- 7.4.32 “GET_MIXER Command”
- 7.4.33 “SET_MATRIX Command”
- 7.4.34 “GET_MATRIX Command”
- 7.4.35 “START_STREAMING Command”
- 7.4.36 “STOP_STREAMING Command”
- 7.4.37 “REGISTER_UNSOLICITED_NOTIFICATION Command”
- 7.4.38 “DEREGISTER_UNSOLICITED_NOTIFICATION Command”
- 7.4.39 “IDENTIFY_NOTIFICATION Unsolicited Response”
- 7.4.40 “GET_AVB_INFO Command”
- 7.4.41 “GET_AS_PATH Command”
- 7.4.42 “GET_COUNTERS Command”
- 7.4.43 “REBOOT Command”
- 7.4.44 “GET_AUDIO_MAP Command”
- 7.4.45 “ADD_AUDIO_MAPPINGS Command”
- 7.4.46 “REMOVE_AUDIO_MAPPINGS Command”
- 7.4.47 “GET_VIDEO_MAP Command”
- 7.4.48 “ADD_VIDEO_MAPPINGS Command”
- 7.4.49 “REMOVE_VIDEO_MAPPINGS Command”
- 7.4.50 “GET_SENSOR_MAP Command”

- 7.4.51 “ADD_SENSOR_MAPPINGS Command”
- 7.4.52 “REMOVE_SENSOR_MAPPINGS Command”
- 7.4.53 “START_OPERATION Command”
- 7.4.54 “ABORT_OPERATION Command”
- 7.4.55 “OPERATION_STATUS Unsolicited Response”
- 7.4.56 “AUTH_ADD_KEY Command”
- 7.4.57 “AUTH_DELETE_KEY Command”
- 7.4.58 “AUTH_GET_KEY_LIST Command”
- 7.4.59 “AUTH_GET_KEY Command”
- 7.4.60 “AUTH_ADD_KEY_TO_CHAIN Command”
- 7.4.61 “AUTH_DELETE_KEY_FROM_CHAIN Command”
- 7.4.62 “AUTH_GET_KEYCHAIN_LIST Command”
- 7.4.63 “AUTH_GET_IDENTITY Command”
- 7.4.64 “AUTH_ADD_TOKEN Command”
- 7.4.65 “AUTH_DELETE_TOKEN Command”
- 7.4.66 “AUTHENTICATE Command”
- 7.4.67 “DEAUTHENTICATE Command”
- 7.4.68 “ENABLE_TRANSPORT_SECURITY Command”
- 7.4.69 “DISABLE_TRANSPORT_SECURITY Command”
- 7.4.70 “ENABLE_STREAM_ENCRYPTION Command”
- 7.4.71 “DISABLE_STREAM_ENCRYPTION Command”
- 7.4.72 “SET_MEMORY_OBJECT_LENGTH Command”
- 7.4.73 “GET_MEMORY_OBJECT_LENGTH Command”
- 7.4.74 “SET_STREAM_BACKUP Command”
- 7.4.75 “GET_STREAM_BACKUP Command”
- 9.2.2.6.1 “Address Access Entity State Machine”
- 9.2.2.9.1 “Legacy A/VC Entity State Machine”
- 9.2.2.12.1 “Vendor Unique Entity State Machine”

5.7 AVDECC Responder

An AVDECC Responder is an AVDECC Entity that does not implement the AVDECC Controller, AVDECC Talker, or AVDECC Listener role but still uses the AVDECC protocols.

5.7.1 Requirements and options

An AVDECC Responder shall use AVDECC messages transported via IEEE 1722 AVTPDUs.

An AVDECC Responder shall implement the following:

- 9.2.1.2 “AVDECC Entity Model format”
- 9.2.2.1 “AVDECC Entity Model Commands”
- 9.2.2.2 “AVDECC Entity Model Responses”
- 9.2.2.3.1 “AVDECC Entity Model Entity State Machine,” implementing the following:
 - 7.4.1 “ACQUIRE_ENTITY Command”
 - 7.4.2 “LOCK_ENTITY Command”
 - 7.4.3 “ENTITY_AVAILABLE Command”
 - 7.4.4 “CONTROLLER_AVAILABLE Command”

An AVDECC Responder may implement the following:

- 9.2.2.3.1 “AVDECC Entity Model Entity State Machine,” implementing any zero or more of the following:
 - 7.4.5 “READ_DESCRIPTOR Command”
 - 7.4.6 “WRITE_DESCRIPTOR Command”
 - 7.4.7 “SET_CONFIGURATION Command”
 - 7.4.8 “GET_CONFIGURATION Command”
 - 7.4.17 “SET_NAME Command”
 - 7.4.18 “GET_NAME Command”
 - 7.4.19 “SET_ASSOCIATION_ID Command”
 - 7.4.20 “GET_ASSOCIATION_ID Command”
 - 7.4.21 “SET_SAMPLING_RATE Command”
 - 7.4.22 “GET_SAMPLING_RATE Command”
 - 7.4.23 “SET_CLOCK_SOURCE Command”
 - 7.4.24 “GET_CLOCK_SOURCE Command”
 - 7.4.25 “SET_CONTROL Command”
 - 7.4.26 “GET_CONTROL Command”
 - 7.4.27 “INCREMENT_CONTROL Command”
 - 7.4.28 “DECREMENT_CONTROL Command”
 - 7.4.29 “SET_SIGNAL_SELECTOR Command”
 - 7.4.30 “GET_SIGNAL_SELECTOR Command”
 - 7.4.31 “SET_MIXER Command”
 - 7.4.32 “GET_MIXER Command”
 - 7.4.33 “SET_MATRIX Command”
 - 7.4.34 “GET_MATRIX Command”
 - 7.4.37 “REGISTER_UNSOLICITED_NOTIFICATION Command”

- 7.4.38 “DEREGISTER_UNSOLICITED_NOTIFICATION Command”
- 7.4.39 “IDENTIFY_NOTIFICATION Unsolicited Response”
- 7.4.40 “GET_AVB_INFO Command”
- 7.4.41 “GET_AS_PATH Command”
- 7.4.42 “GET_COUNTERS Command”
- 7.4.43 “REBOOT Command”
- 7.4.53 “START_OPERATION Command”
- 7.4.54 “ABORT_OPERATION Command”
- 7.4.55 “OPERATION_STATUS Unsolicited Response”
- 7.4.56 “AUTH_ADD_KEY Command”
- 7.4.57 “AUTH_DELETE_KEY Command”
- 7.4.58 “AUTH_GET_KEY_LIST Command”
- 7.4.59 “AUTH_GET_KEY Command”
- 7.4.60 “AUTH_ADD_KEY_TO_CHAIN Command”
- 7.4.61 “AUTH_DELETE_KEY_FROM_CHAIN Command”
- 7.4.62 “AUTH_GET_KEYCHAIN_LIST Command”
- 7.4.63 “AUTH_GET_IDENTITY Command”
- 7.4.64 “AUTH_ADD_TOKEN Command”
- 7.4.65 “AUTH_DELETE_TOKEN Command”
- 7.4.66 “AUTHENTICATE Command”
- 7.4.67 “DEAUTHENTICATE Command”
- 7.4.68 “ENABLE_TRANSPORT_SECURITY Command”
- 7.4.69 “DISABLE_TRANSPORT_SECURITY Command”
- 7.4.72 “SET_MEMORY_OBJECT_LENGTH Command”
- 7.4.73 “GET_MEMORY_OBJECT_LENGTH Command”
- 7.4.74 “SET_STREAM_BACKUP Command”
- 7.4.75 “GET_STREAM_BACKUP Command”
- 9.2.2.6.1 “Address Access Entity State Machine”
- 9.2.2.9.1 “Legacy A/VC Entity State Machine”
- 9.2.2.12.1 “Vendor Unique Entity State Machine”

5.8 AVDECC Proxy Server requirements and options

An AVDECC Proxy Server (APS, see Annex C) shall implement the following:

- C.2 “DNS-SD Service Name”
- C.3 “DNS-SD TXT Record”

— C.5.2 “APS State Machine”

5.9 AVDECC Proxy Client requirements and options

An AVDECC Proxy Client (APC, see Annex C) shall implement the following:

— C.5.3 “APC State Machine”

6. AVDECC Entity Discovery

6.1 Overview

AVDECC Entity discovery is the process in which AVDECC Controllers identify all of the AVDECC Entities currently available on a network and identify them as they are added to or removed from the network. The AVDECC Discovery Protocol (ADP) is used for this purpose.

6.2 AVDECC Discovery Protocol

ADP is a protocol based on IEEE Std 1722-2011 control AVTPDUs allowing AVDECC Entities to be discovered by each other. ADP uses three message types to do the following:

- a) Announce that an AVDECC Entity is available.
- b) Announce that an AVDECC Entity is departing.
- c) Discover one or all of the AVDECC Entities on the network.

6.2.1 AVDECC Discovery Protocol Data Unit format

The AVDECC Discovery Protocol Data Unit (ADPDU) follows the IEEE Std 1722-2011 control AVTPDU header.

The ADPDU contains the following fields:

- **entity_model_id:** 64 bits
- **entity_capabilities:** 32 bits
- **talker_stream_sources:** 16 bits
- **talker_capabilities:** 16 bits
- **listener_stream_sinks:** 16 bits
- **listener_capabilities:** 16 bits
- **controller_capabilities:** 32 bits
- **available_index:** 32 bits
- **gptp_grandmaster_id:** 64 bits
- **gptp_domain_number:** 8 bits

- **reserved0**: 24 bits
- **identify_control_index**: 16 bits
- **interface_index**: 16 bits
- **association_id**: 64 bits
- **reserved1**: 32 bits

Figure 6.1 shows these fields with offset zero (0) shown as the first octet of the ADPDU following the IEEE Std 1722-2011 control AVTPDU header.

The ADP redefines several fields within the IEEE Std 1722-2011 control AVTPDU header. The following fields are redefined:

- **control_data** is redefined as **message_type**
- **status** is redefined as **valid_time**
- **stream_id** is redefined as **entity_id**

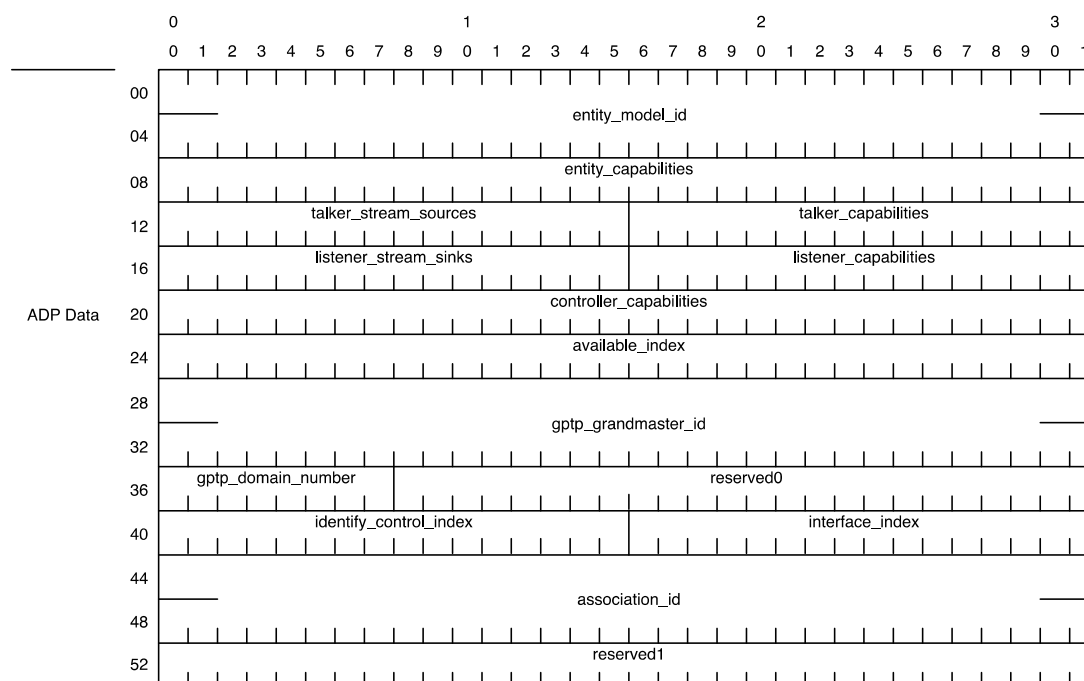


Figure 6.1—ADPDU format

6.2.1.1 cd field

The IEEE Std 1722-2011 control AVTPDU **cd** (control/data) bit is set to one (1).

6.2.1.2 subtype field

The IEEE Std 1722-2011 control AVTPDU subtype is set to the ADP subtype, 7a₁₆.

6.2.1.3 sv field

The IEEE Std 1722-2011 control AVTPDU **sv** (stream valid) bit is set to zero (0).

6.2.1.4 version field

The IEEE Std 1722-2011 control AVTPDU **version** field is set to the version of IEEE Std 1722 being used.

6.2.1.5 message_type (control_data) field

The IEEE Std 1722-2011 control AVTPDU **control_data** field is renamed to **message_type** in ADP. The **message_type** is set to one of the defined ADP message types as defined in Table 6.1.

Table 6.1—message_type field

Value	Function	Meaning
0	ENTITY_AVAILABLE	The AVDECC Entity is available.
1	ENTITY_DEPARTING	The AVDECC Entity is going away.
2	ENTITY_DISCOVER	Request for AVDECC Entities on the network to send an ENTITY_AVAILABLE message if the requested Entity ID is zero (0) or is the AVDECC Entities' Entity ID.
3 to 15	—	Reserved for future use.

6.2.1.6 valid_time (status) field

The IEEE Std 1722-2011 control AVTPDU **status** field is renamed to **valid_time** in ADP. This field indicates how long the record will be valid for in two-second increments.

For ENTITY_AVAILABLE messages, this field is set to the AVDECC Entities' valid time period.

The valid time period for an AVDECC Entity is between 2 and 62 seconds, resulting in a **valid_time** field value of one 1 to 31. The valid time is a tradeoff between the amount of traffic generated as a result of advertisements and the time it takes for an AVDECC Entity to be perceived to be missing from the network.

The default valid time is 62 seconds, which results in an ENTITY_AVAILABLE message being sent every 15 seconds to 16 seconds, and taking up to 62 seconds to notice that an AVDECC Entity has been removed from the network.

For all other messages, this field is set to zero (0).

6.2.1.7 control_data_length field

The IEEE Std 1722-2011 control AVTPDU **control_data_length** field for ADP is set to 56.

6.2.1.8 entity_id (stream_id) field

The IEEE Std 1722-2011 control AVTPDU **stream_id** field is renamed to **entity_id** in ADP. It carries the EUI-64 identifier of the AVDECC Entity.

The **entity_id** is set to an EUI-64 uniquely identifying the AVDECC Entity. The EUI-64 is assigned by the vendor manufacturing the device. It can be the EUI-64 computed from the AVDECC Entity's primary network port's MAC address.

An AVDECC Entity shall have only one Entity ID for use with the AVDECC protocols.

All AVDECC Entities shall have a unique **entity_id** value.

NOTE—In the case of an End Station containing multiple AVDECC Entities, each AVDECC Entity has a unique Entity ID.²⁰

6.2.1.9 entity_model_id field

The **entity_model_id** field is an EUI-64 used to identify an AVDECC Entity data model from a vendor. These numbers are vendor-specific and different AVDECC Entity data models shall have different **entity_model_id** values.

If a firmware revision changes the structure of an AVDECC Entity data model then it shall use a new unique **entity_model_id**.

The structure of an AVDECC Entity model is considered changed if any descriptor fields in the AVDECC Entity are different excluding the following fields:

- In all descriptors: **object_name**
- In the ENTITY descriptor: **available_index**, **association_id**, **entity_name**, **firmware_version**, **group_name**, **serial_number**, and **current_configuration**
- In AUDIO_UNIT descriptors: **current_sampling_rate**
- In STREAM_INPUT and STREAM_OUTPUT descriptors: **current_format**
- In CLOCK_SOURCE descriptors: **clock_source_identifier** and **clock_source_flags**
- In CLOCK_DOMAIN descriptors: **clock_source_index**
- In CONTROL, MIXER, MATRIX or SIGNAL_TRANSCODER descriptors with a **control_value_type** of CONTROL_LINEAR_UINT8 to CONTROL_LINEAR_DOUBLE: **current[X]** subfields of the **value_details**
- In CONTROL, MATRIX or SIGNAL_TRANSCODER descriptors with a **control_value_type** of CONTROL_SELECTOR_UINT8 to CONTROL_SELECTOR_STRING: **current** subfield of the **value_details**
- In CONTROL descriptors with a **control_value_type** of CONTROL_BODE_PLOT: **current_frequency[X]**, **current_magnitude[X]** and **current_phase[X]** subfields of the **value_details**
- In CONTROL, MATRIX or SIGNAL_TRANSCODER descriptors with a **control_value_type** of CONTROL_ARRAY_UINT8 to CONTROL_ARRAY_DOUBLE: **current[X]** subfields of the **value_details**
- In CONTROL descriptors with a **control_value_type** of CONTROL_UTF8, CONTROL_SMPTE_TIME, CONTROL_SAMPLE_RATE, CONTROL_GPTP_TIME or CONTROL_VENDOR: **value_details**
- In SIGNAL_SELECTOR descriptors: **current_signal_type**, **current_signal_index**, and **current_signal_output**

²⁰ Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

- In VIDEO_CLUSTER descriptors: **current_format_specific**, **current_sampling_rate**, **current_aspect_ratio**, **current_size**, and **current_color_space**
- In SENSOR_CLUSTER descriptors: **current_format** and **current_sampling_rate**
- In MEMORY_OBJECT descriptors: **length**

NOTE—The **entity_model_id** is not a device's product or model number.

6.2.1.10 entity_capabilities field

The **entity_capabilities** field is used to identify support for various protocols and features on the AVDECC Entity. This field is set to a combination of the valid bits as defined by Table 6.2.

Table 6.2—entity_capabilities field

Bit	Field Value	Function	Meaning
31	00000001 ₁₆	EFU_MODE	Entity Firmware Upgrade mode is enabled on the AVDECC Entity. When this flag is set, the AVDECC Entity is in the mode to perform an AVDECC Entity firmware upgrade.
30	00000002 ₁₆	ADDRESS_ACCESS_SUPPORTED	Supports receiving the ADDRESS_ACCESS commands as defined in 9.2.1.3.
29	00000004 ₁₆	GATEWAY_ENTITY	AVDECC Entity serves as a gateway to a device on another type of media (typically a IEEE Std 1394 device) by proxying control services for it.
28	00000008 ₁₆	AEM_SUPPORTED	Supports receiving the AVDECC Entity Model (AEM) AECp commands as defined in 9.2.1.2.
27	00000010 ₁₆	LEGACY_AVC	Supports using IEEE Std 1394 AV/C protocol (e.g., for a IEEE Std 1394 devices through a gateway).
26	00000020 ₁₆	ASSOCIATION_ID_SUPPORTED	The AVDECC Entity supports the use of the association_id field for associating the AVDECC Entity with other AVDECC entities.
25	00000040 ₁₆	ASSOCIATION_ID_VALID	The association_id field contains a valid value. This bit shall only be set in conjunction with ASSOCIATION_ID_SUPPORTED.
24	00000080 ₁₆	VENDOR_UNIQUE_SUPPORTED	Supports receiving the AEM VENDOR_UNIQUE commands as defined in 9.2.1.5.
23	00000100 ₁₆	CLASS_A_SUPPORTED	Supports sending and/or receiving Class A Streams.
22	00000200 ₁₆	CLASS_B_SUPPORTED	Supports sending and/or receiving Class B Streams.
21	00000400 ₁₆	GPTP_SUPPORTED	The AVDECC Entity implements IEEE Std 802.1AS-2011.

Table 6.2—entity_capabilities field (continued)

Bit	Field Value	Function	Meaning
20	00000800 ₁₆	AEM_AUTHENTICATION_SUPPORTED	Supports using AEM Authentication via the AUTHENTICATE command as defined in 7.4.66. This flag shall only be set if the AEM_SUPPORTED flag is set.
19	00001000 ₁₆	AEM_AUTHENTICATION_REQUIRED	Requires the use of AEM Authentication via the AUTHENTICATE command as defined in 7.4.66. This flag shall only be set if the AEM_SUPPORTED flag is set.
18	00002000 ₁₆	AEM_PERSISTENT_ACQUIRE_SUPPORTED	Supports the use of the PERSISTENT flag in the ACQUIRE command as defined in 7.4.1. This flag shall only be set if the AEM_SUPPORTED flag is set.
17	00004000 ₁₆	AEM_IDENTIFY_CONTROL_INDEX_VALID	The identify_control_index field contains a valid index of an AEM CONTROL descriptor for the primary IDENTIFY Control in the current Configuration. This flag shall only be set if the AEM_SUPPORTED flag is set.
16	00008000 ₁₆	AEM_INTERFACE_INDEX_VALID	The interface_index field contains a valid index of an AEM AVB_INTERFACE descriptor for interface in the current Configuration which is transmitting the ADPDU. This flag shall only be set if the AEM_SUPPORTED flag is set.
15	00010000 ₁₆	GENERAL_CONTROLLER_IGNORE	General purpose AVDECC Controllers ignore the presence of this AVDECC Entity when this flag is set.
14	00020000 ₁₆	ENTITY_NOT_READY	The AVDECC Entity is not ready to be enumerated or connected by an AVDECC Controller.
0 to 13	—	—	Reserved for future use.

6.2.1.11 talker_stream_sources field

The **talker_stream_sources** field is used to identify the number of Streams an AVDECC Talker is capable of sourcing simultaneously.

6.2.1.12 talker_capabilities field

The **talker_capabilities** field is a 16-bit bitfield used to identify the capabilities of an AVDECC Talker. This field is set to a combination of the valid bits as defined by Table 6.3.

Table 6.3—talker_capabilities field

Bit	Field Value	Function	Meaning
15	0001 ₁₆	IMPLEMENTED	Implements an AVDECC Talker.

Table 6.3—talker_capabilities field (continued)

Bit	Field Value	Function	Meaning
7 to 14	—	—	Reserved for future use.
6	0200 ₁₆	OTHER_SOURCE	The AVDECC Talker has other Stream sources not covered by the following.
5	0400 ₁₆	CONTROL_SOURCE	The AVDECC Talker has Control Stream sources.
4	0800 ₁₆	MEDIA_CLOCK_SOURCE	The AVDECC Talker has Media Clock Stream sources.
3	1000 ₁₆	SMPTE_SOURCE	The AVDECC Talker has Society of Motion Picture and Television Engineers (SMPTE) time code Stream sources.
2	2000 ₁₆	MIDI_SOURCE	The AVDECC Talker has Musical Instrument Digital Interface (MIDI) Stream sources.
1	4000 ₁₆	AUDIO_SOURCE	The AVDECC Talker has Audio Stream sources.
0	8000 ₁₆	VIDEO_SOURCE	The AVDECC Talker has Video Stream sources (which can include embedded audio).

6.2.1.13 listener_stream_sinks field

The **listener_stream_sinks** field is used to identify the number of Streams an AVDECC Listener is capable of sinking simultaneously.

6.2.1.14 listener_capabilities field

The **listener_capabilities** field is a 16-bit bitfield used to identify the capabilities of an AVDECC Listener. This field is set to a combination of the valid bits as defined by Table 6.4.

Table 6.4—Listener capabilities field

Bit	Field Value	Function	Meaning
15	0001 ₁₆	IMPLEMENTED	Implements an AVDECC Listener.
7 to 14	—	—	Reserved for future use.
6	0200 ₁₆	OTHER_SINK	The AVDECC Listener has other Stream sinks not covered by the following.
5	0400 ₁₆	CONTROL_SINK	The AVDECC Listener has Control Stream sinks.
4	0800 ₁₆	MEDIA_CLOCK_SINK	The AVDECC Listener has Media Clock Stream sinks.
3	1000 ₁₆	SMPTE_SINK	The AVDECC Listener has SMPTE time code Stream sinks.
2	2000 ₁₆	MIDI_SINK	The AVDECC Listener has MIDI Stream sinks.
1	4000 ₁₆	AUDIO_SINK	The AVDECC Listener has Audio Stream sinks.
0	8000 ₁₆	VIDEO_SINK	The AVDECC Listener has Video Stream sinks (which can include embedded audio).

6.2.1.15 controller_capabilities field

The **controller_capabilities** field is used to identify the AVDECC Controller capabilities of an AVDECC Entity. This is a bit field, with the lowest bit indicating that the AVDECC Entity implements an AVDECC Controller. This field is set to a combination of the valid bits as defined by Table 6.5.

Table 6.5—controller_capabilities field

Bit	Field Value	Function	Meaning
31	00000001 ₁₆	IMPLEMENTED	Implements an AVDECC Controller.
0 to 20	—	—	Reserved for future use.

6.2.1.16 available_index field

The **available_index** field is used to differentiate between availability cycles of the AVDECC Entity. The **available_index** value is incremented after transmitting an ENTITY_AVAILABLE message and is reset to zero (0) when transmitting an ENTITY_DEPARTING or after a power cycle.

6.2.1.17 gtp_grandmaster_id field

The **gtp_grandmaster_id** field is used to differentiate between devices in different gPTP domains. The value of **gtp_grandmaster_id** is per interface.

If the GPTP_SUPPORTED flag is set in **entity_capabilities** then the **gtp_grandmaster_id** field is set to the ClockIdentity of the grandmaster in the gPTP domain that this Entity is participating in on the interface transmitting the ADPDU.

If the GPTP_SUPPORTED flag is not set in **entity_capabilities** then the **gtp_grandmaster_id** field is set to zero (0) on transmit and ignored on receive.

6.2.1.18 gtp_domain_number field

The **gtp_domain_number** field is used to differentiate between devices in different gPTP domains. The value of **gtp_domain_number** is per interface.

If the GPTP_SUPPORTED flag is set in **entity_capabilities** then the **gtp_domain_number** field is set to the domainNumber of the grandmaster in the gPTP domain that this Entity is participating in on the interface transmitting the ADPDU.

If the GPTP_SUPPORTED flag is not set in **entity_capabilities** then the **gtp_domain_number** field is set to zero (0) on transmit and ignored on receive.

6.2.1.19 identify_control_index

The **identify_control_index** field is used to advertise the appropriate AEM CONTROL descriptor (see 7.2.22) that is the primary IDENTIFY Control (see 7.3.4.2) for the AVDECC Entity.

If the AEM_IDENTIFY_CONTROL_INDEX_VALID flag is set in **entity_capabilities** then the **identify_control_index** field is set to a CONTROL descriptor index that can be used with the SET_CONTROL AEM command (see 7.4.25).

If the `AEM_IDENTIFY_CONTROL_INDEX_VALID` flag is not set in **entity_capabilities** then the **identify_control_index** field is set to zero (0) on transmit and ignored on receive.

6.2.1.20 interface_index

The **interface_index** field is used to advertise the AEM AVB_INTERFACE descriptor (see 7.2.8) that represents the interface that is transmitting this ADPDU.

If the `AEM_INTERFACE_INDEX_VALID` flag is set in **entity_capabilities** then the **interface_index** field is set to the AVB_INTERFACE descriptor index of the interface transmitting the ADPDU.

If the `AEM_INTERFACE_INDEX_VALID` flag is not set in **entity_capabilities** then the **interface_index** field is set to zero (0) on transmit and ignored on receive.

6.2.1.21 association_id

The **association_id** field is used to associate multiple AVDECC Entities into a logical collection. This allows each loudspeaker of a multi-channel rig to be a separate AVDECC Entity but to be associated by the AVDECC Controller into a single logical AVDECC Entity.

If the `ASSOCIATION_ID_SUPPORTED` flag is not set in the **entity_capabilities** field then this field is set to zero (0).

If `ASSOCIATION_ID_SUPPORTED` flag is set and `ASSOCIATION_ID_VALID` flag is not set in the **entity_capabilities** field then this field is set to zero (0).

If `ASSOCIATION_ID_SUPPORTED` and `ASSOCIATION_ID_VALID` flags are set in the **entity_capabilities** field then the value of this field is the EUI-64 used to associate the AVDECC Entities (all associated AVDECC Entities will have the same EUI-64 in this field).

6.2.2 Protocol specification

All ADPDUs are transmitted to the ADP multicast destination MAC address defined in Table B.1

The ADP protocol is implemented through three state machines, an Advertising Entity State machine for each AVDECC Entity being published on the End Station, an Advertising Interface State machine for each AVB Interface of the AVDECC Entity being published in the End Station, and a Discovery State machine for each AVDECC Entity implementing an AVDECC Controller or requiring Entity discovery.

There is one instance of the Discovery State machine for each AVB Interface of the AVDECC Entity implementing discovery. The AVDECC Entity may provide its own coordination between the discovery state machines to track AVDECC Entities which appear on multiple interfaces.

6.2.3 Global state machine variables

6.2.3.1 currentTime

The **currentTime** global variable contains the current time of a local clock, which always advances forward.

6.2.3.2 entityInfo

The entityInfo variable is a structure containing all of the common information required to populate the fields of an ADPDU across all of the interfaces. This variable is only used for the Advertising Entity and Advertising Interface state machines.

6.2.4 Advertise Entity State Machine

6.2.4.1 State machine variables

6.2.4.1.1 reannounceTimerTimeout

The reannounceTimerTimeout is the time relative to currentTime that the state machine is required to send an ENTITY_AVAILABLE message.

6.2.4.1.2 needsAdvertise

The needsAdvertise variable is a Boolean used by the Advertising Interface State machine to signal the Advertising Entity State machine that it needs to transmit an ENTITY_AVAILABLE message on all interfaces.

6.2.4.1.3 doTerminate

doTerminate is a Boolean indicating that a request has been made to the Advertise Entity State machine to terminate advertising and that the state machine is to be ended.

6.2.4.2 State machine functions

6.2.4.2.1 sendAvailable()

The sendAvailable function sets all of the **doAdvertise** booleans on all of the Advertising Interface state machines to signal them to transmit an advertise message.

6.2.4.3 State machine diagram

Figure 6.2 shows the Advertise Entity State Machine diagram.

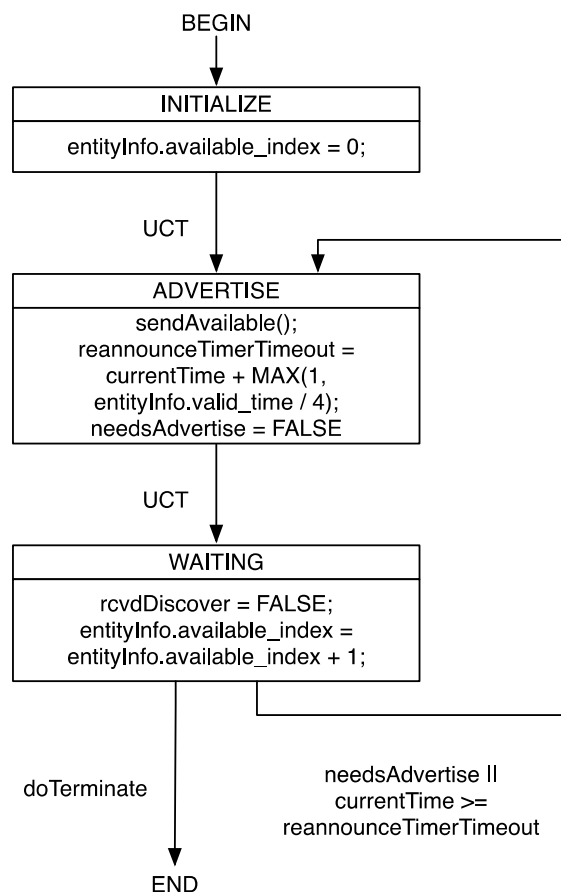


Figure 6.2—Advertise Entity State Machine

6.2.5 Advertise Interface State Machine

6.2.5.1 State machine variables

6.2.5.1.1 advertisedGrandmasterID

The advertisedGrandmasterID is the ClockIdentity of the grandmaster in the gPTP domain that this interface is participating in.

6.2.5.1.2 entityInfo

The entityInfo variable is a structure containing all of the information required to populate the fields of an ADPDU.

6.2.5.1.3 rcvdDiscover

The rcvdDiscover variable is a Boolean indicating that an ADPDU has been received with a message_type of ENTITY_DISCOVER.

6.2.5.1.4 entityID

The entityID variable is an unsigned 64-bit value containing the **entity_id** from the received ENTITY_DISCOVER ADPDU. This is set at the same time from the same ADPDU as rcvdDiscover.

6.2.5.1.5 doTerminate

doTerminate is a Boolean indicating that a request has been made to the Advertise Interface State machine to terminate advertising and that the state machine is to be ended.

6.2.5.1.6 doAdvertise

doAdvertise is a Boolean indicating that a request has been made to the Advertise Interface State machine to transmit an ENTITY_AVAILABLE ADPDU by the Advertise Entity state machine.

6.2.5.1.7 linkIsUp

The linkIsUp variable is a Boolean indicating that the interface's media link is up (connected). This variable is updated whenever the state of the link changes.

6.2.5.1.8 lastLinkIsUp

The lastLinkIsUp variable is a Boolean indicating the last known value of linkIsUp; this variable is used to determine when the value of linkIsUp changes.

6.2.5.2 State machine functions

6.2.5.2.1 txEntityAvailable()

The txEntityAvailable function transmits an ENTITY_AVAILABLE message.

If the AVDECC Entity has more than one enabled network port, then the same ADPDU is sent out each port.

The **cd**, **sv**, **version**, and **control_data_length** fields are set as per 6.2.1.

The **message_type** field is set to ENTITY_AVAILABLE.

All other fields are set per the entityInfo variable of the state machine.